



APP.4.3 Relationale Datenbanken

1. Beschreibung

1.1. Einleitung

Datenbanksysteme (DBS) sind ein oft genutztes Hilfsmittel, um IT-gestützt große Datensammlungen zu organisieren, zu erzeugen, zu verändern und zu verwalten. Ein DBS besteht aus dem so genannten Datenbankmanagementsystem (DBMS) und einer oder mehreren Datenbanken. Eine Datenbank ist eine Zusammenstellung von Daten samt ihrer Beschreibung (Metadaten), die dauerhaft im Datenbanksystem abgelegt werden. Da Datenbanksysteme oft eine zentrale Bedeutung in einer IT-Infrastruktur einnehmen, ergeben sich an sie wesentliche Sicherheitsanforderungen. Meist sind Kernprozesse einer Institution von den Informationen aus den Datenbanken abhängig. Dadurch ergeben sich entsprechende Verfügbarkeitsanforderungen. Zusätzlich bestehen oft hohe Anforderungen an die Vertraulichkeit und Integrität der in den Datenbanken gespeicherten Informationen.

1.2. Zielsetzung

Ziel dieses Bausteins ist es, relationale Datenbanksysteme sicher betreiben zu können sowie die Informationen, die in Datenbanken verarbeitet und gespeichert werden, angemessen zu schützen. Dazu werden Anforderungen beschrieben, mit denen sich Datenbanksysteme sicher planen, umsetzen und betreiben lassen und durch die Gefährdungen reduziert werden können.

1.3. Abgrenzung und Modellierung

Der Baustein APP.4.3 *Relationale Datenbanken* ist auf jedes relationale Datenbanksystem einmal anzuwenden.

In diesem Baustein werden Anforderungen an relationale Datenbanksysteme beschrieben. Sicherheitsanforderungen an nicht-relationale Datenbanksysteme sind nicht Gegenstand des vorliegenden Bausteins.

Um die Informationen in den Datenbanken durchgängig zu schützen, sollten bereits in der Anwendungsentwicklung Sicherheitsanforderungen an den Aufbau der Datenbanktabellen und den Zugriff auf die Datenbank beachtet werden. Anforderungen zu diesen Themen werden jedoch nicht in diesem Baustein aufgeführt.

Ebenso geht der Baustein nicht auf Gefährdungen und Anforderungen ein, die das Betriebssystem und die Hardware betreffen, auf denen das Datenbanksystem installiert ist. Aspekte dazu finden sich in den entsprechenden betriebssystemspezifischen Bausteinen der Schicht SYS *IT-Systeme*, z. B. SYS.1.3 *Server unter Linux und Unix* oder SYS.1.2.3 *Windows Server*.

Relationale Datenbanksysteme sollten grundsätzlich im Rahmen der Bausteine OPR.4 *Identitäts- und Berechtigungsmanagement*, OPS.1.1.3 *Patch- und Änderungsmanagement*, CON.3 *Datensicherungskonzept*, OPS.1.2.2 *Archivierung*, OPS.1.1.5 *Protokollierung* sowie OPS.1.1.2 *Ordnungsgemäße IT-Administration* mit berücksichtigt werden.

2. Gefährdungslage

Da IT-Grundschutz-Bausteine nicht auf individuelle Informationsverbünde eingehen können, werden zur Darstellung der Gefährdungslage typische Szenarien zugrunde gelegt. Die folgenden spezifischen Bedrohungen und Schwachstellen sind für den Baustein APP.4.3 *Relationale Datenbanken* von besonderer Bedeutung.

2.1. Unzureichende Dimensionierung der Systemressourcen

Verfügt die Hardware eines Datenbanksystems nicht über genügend Systemressourcen, kann die Datenbank ganz ausfallen oder fehlerhaft arbeiten. Dadurch können beispielsweise Daten nicht gespeichert werden. Auch können zu Stoßzeiten die Ressourcen stark ausgelastet werden. Dadurch kann sich die Performance verschlechtern. Dies wiederum kann dazu führen, dass Anwendungen nicht oder nicht fehlerfrei ausgeführt werden.

2.2. Aktivierte Standard-Konten

Bei der Erstinstallation bzw. im Auslieferungszustand eines Datenbankmanagementsystems sind Standard-Konten (Konten der Benutzenden und Administrierenden) häufig nicht oder nur mit Passwörtern gesichert, die öffentlich bekannt sind. Dadurch kann es passieren, dass diese Konten missbräuchlich genutzt werden. Beispielsweise können sich Angreifende mit den öffentlich bekannten Anmeldedaten am Datenbankmanagementsystem als Benutzende oder sogar als Administrierende anmelden. Danach können sie die Konfiguration oder die gespeicherten Daten auslesen, manipulieren oder löschen.

2.3. Unverschlüsselte Datenbankanbindung

In der Standardkonfiguration verbinden sich viele Datenbankmanagementsysteme unverschlüsselt mit den Anwendungen. Wird zwischen Anwendungen und Datenbankmanagementsystem unverschlüsselt kommuniziert, können übertragene Daten und Zugangsinformationen mitgelesen oder auf dem Transportweg manipuliert werden.

2.4. Datenverlust in der Datenbank

Durch Hardware- oder Softwarefehler sowie durch menschliches Versagen können Daten in der Datenbank verloren gehen. Da in Datenbanken meist wichtige Informationen für Anwendungen gespeichert sind, können Dienste ausfallen oder ganze Produktionsprozesse stillstehen.

2.5. Integritätsverlust der gespeicherten Daten

Durch falsch konfigurierte Datenbanken, Softwarefehler oder manipulierte Daten kann die Integrität der Informationen in der Datenbank verletzt werden. Wird dies nicht oder erst spät bemerkt, können Kernprozesse der Institution stark beeinträchtigt werden. Werden beispielsweise die Integritätsbeziehungen (referenzielle Integrität) zwischen den Tabellen nicht korrekt definiert, kann dies dazu führen, dass die Daten in der Datenbank fehlerhaft sind. Wird dieser Fehler erst im Produktivbetrieb oder gar nicht bemerkt, müssen nicht nur die inkonsistenten Daten aufwändig bereinigt und rekonstruiert werden. Es kann mit der Zeit auch ein großer Schaden entstanden sein, beispielsweise wenn es sich um kritische Daten, zum Beispiel steuerrelevante Daten, Rechnungsdaten oder gar um Steuerungsdaten für ganze Produktionssysteme handelt.

2.6. SQL-Injections

Eine häufige Angriffsmethode auf Datenbanksysteme sind SQL-Injections. Greift eine Anwendung auf die Daten einer SQL-Datenbank zu, so werden Befehle in Form von SQL-Anweisungen an das DBMS übermittelt. Werden Eingabedaten innerhalb der Anwendung unzureichend validiert, können Angreifende eigene SQL-Befehle in die Anwendung einschleusen, die dann mit der Berechtigung des Dienstkontos der Anwendung bearbeitet werden. Angreifende können so Daten lesen, manipulieren, löschen, neue Daten hinzufügen oder auch Systembefehle aufrufen. Obwohl SQL-Injections primär die Anwendungen im Frontend betreffen, wirken sie sich auch erheblich auf das Datenbanksystem selbst und die damit verbundene Infrastruktur aus.

2.7. Unsichere Konfiguration des Datenbankmanagementsystems

Häufig sind in der Standardkonfiguration des Datenbankmanagementsystems nicht benötigte Funktionen aktiviert, die es bei einem potenziellen Angriff erleichtern, Informationen aus der Datenbank auszulesen oder zu manipulieren. Beispielsweise können sich Angreifende aufgrund einer unveränderten Standardinstallation mit einer von der Institution nicht benutzten Programmierschnittstelle verbinden, um das DBMS zu administrieren, ohne sich dafür authentifizieren zu müssen. Dadurch können sie unerlaubt auf die Datenbanken der Institution zugreifen.

2.8. Malware und unsichere Datenbank-Skripte

Bei vielen Datenbankmanagementsystemen ist es möglich, bestimmte Aktionen über Skripte zu automatisieren, die im Kontext der Datenbank ausgeführt werden, z. B. mithilfe der Procedural Language/Structured Query Language (PL/SQL). Dazu gehören unter anderem auch sogenannte Datenbanktrigger. Werden diese jedoch von den Zuständigen ungeprüft benutzt, könnten die Datenbank-Skripte nicht den Anforderungen an die Softwareentwicklung der Institution genügen.

Ebenfalls können bei Angriffen Kernfunktionen einer Datenbank, wie z. B. Data Dictionary Tables manipuliert werden, beispielsweise mithilfe von Schadprogrammen oder Datenbank-Skripten. Diese Art von Angriffen ist nur schwer zu entdecken. Qualitätsmängel in diesen Skripten und Malware können sowohl die Vertraulichkeit als auch die Integrität und die Verfügbarkeit der in den Datenbanken abgelegten Daten gefährden.

3. Anforderungen

Im Folgenden sind die spezifischen Anforderungen des Bausteins APP.4.3 *Relationale Datenbanken* aufgeführt. Der oder die Informationssicherheitsbeauftragte (ISB) ist dafür zuständig, dass alle Anforderungen gemäß dem festgelegten Sicherheitskonzept erfüllt und überprüft werden. Bei strategischen Entscheidungen ist der oder die ISB stets einzubeziehen.

Im IT-Grundschutz-Kompendium sind darüber hinaus weitere Rollen definiert. Sie sollten besetzt werden, insofern dies sinnvoll und angemessen ist.

Zuständigkeiten	Rollen
Grundsätzlich zuständig	IT-Betrieb
Weitere Zuständigkeiten	Fachverantwortliche, Entwickelnde

Genau eine Rolle sollte *Grundsätzlich zuständig* sein. Darüber hinaus kann es noch *Weitere Zuständigkeiten* geben. Falls eine dieser weiteren Rollen für die Erfüllung einer Anforderung vorrangig zuständig ist, dann wird diese Rolle hinter der Überschrift der Anforderung in eckigen Klammern aufgeführt. Die Verwendung des Singulärs oder Plurals sagt nichts darüber aus, wie viele Personen diese Rollen ausfüllen sollen.

3.1. Basis-Anforderungen

Die folgenden Anforderungen MÜSSEN für diesen Baustein vorrangig erfüllt werden.

APP.4.3.A1 Erstellung einer Sicherheitsrichtlinie für Datenbanksysteme (B)

Ausgehend von der allgemeinen Sicherheitsrichtlinie der Institution MUSS eine spezifische Sicherheitsrichtlinie für Datenbanksysteme erstellt werden. Darin MÜSSEN nachvollziehbar Anforderungen und Vorgaben beschrieben sein, wie Datenbanksysteme sicher betrieben werden sollen. Die Richtlinie MUSS allen im Bereich Datenbanksysteme zuständigen Mitarbeitenden bekannt sein. Sie MUSS grundlegend für ihre Arbeit sein. Wird die Richtlinie verändert oder wird von den Anforderungen abgewichen, MUSS dies mit dem oder der ISB abgestimmt und dokumentiert werden. Es MUSS regelmäßig überprüft werden, ob die Richtlinie noch korrekt umgesetzt ist. Die Ergebnisse MÜSSEN sinnvoll dokumentiert werden.

APP.4.3.A2 ENTFALLEN (B)

Diese Anforderung ist entfallen.

APP.4.3.A3 Basishärtung des Datenbankmanagementsystems (B)

Das Datenbankmanagementsystem MUSS gehärtet werden. Hierfür MUSS eine Checkliste mit den durchzuführenden Schritten zusammengestellt und abgearbeitet werden. Passwörter DÜRFEN NICHT im Klartext gespeichert werden. Die Basishärtung MUSS regelmäßig überprüft und, falls erforderlich, angepasst werden.

APP.4.3.A4 Geregeltes Anlegen neuer Datenbanken (B)

Neue Datenbanken MÜSSEN nach einem definierten Prozess angelegt werden. Wenn eine neue Datenbank angelegt wird, MÜSSEN Grundinformationen zur Datenbank nachvollziehbar dokumentiert werden.

APP.4.3.A5 ENTFALLEN (B)

Diese Anforderung ist entfallen.

APP.4.3.A6 ENTFALLEN (B)

Diese Anforderung ist entfallen.

APP.4.3.A7 ENTFALLEN (B)

Diese Anforderung ist entfallen.

APP.4.3.A8 ENTFALLEN (B)

Diese Anforderung ist entfallen.

APP.4.3.A9 Datensicherung eines Datenbanksystems (B)

Es MÜSSEN regelmäßig Systemsicherungen des DBMS und der Daten durchgeführt werden. Auch bevor eine Datenbank neu erzeugt wird, MUSS das Datenbanksystem gesichert werden. Hierfür SOLLTEN die dafür zulässigen Dienstprogramme benutzt werden.

Alle Transaktionen SOLLTEN so gesichert werden, dass sie jederzeit wiederherstellbar sind. Wenn die Datensicherung die verfügbaren Kapazitäten übersteigt, SOLLTE ein erweitertes Konzept erstellt werden, um die Datenbank zu sichern, z. B. eine inkrementelle Sicherung. Abhängig vom Schutzbedarf der Daten SOLLTEN die Wiederherstellungsparameter vorgegeben werden (siehe CON.3 *Datensicherungskonzept*).

3.2. Standard-Anforderungen

Gemeinsam mit den Basis-Anforderungen entsprechen die folgenden Anforderungen dem Stand der Technik für diesen Baustein. Sie SOLLTEN grundsätzlich erfüllt werden.

APP.4.3.A10 ENTFALLEN (S)

Diese Anforderung ist entfallen.

APP.4.3.A11 Ausreichende Dimensionierung der Hardware (S) [Fachverantwortliche]

Datenbankmanagementsysteme SOLLTEN auf ausreichend dimensionierter Hardware installiert werden. Die Hardware SOLLTE über genügend Reserven verfügen, um auch eventuell steigenden Anforderungen gerecht zu werden. Zeichnen sich trotzdem während des Betriebs Ressourcenengpässe ab, SOLLTEN diese frühzeitig behoben werden. Wenn die Hardware dimensioniert wird, SOLLTE das erwartete Wachstum für den geplanten Einsatzzeitraum berücksichtigt werden.

APP.4.3.A12 Einheitlicher Konfigurationsstandard von Datenbankmanagementsystemen (S)

Für alle eingesetzten Datenbankmanagementsysteme SOLLTE ein einheitlicher Konfigurationsstandard definiert werden. Alle Datenbankmanagementsysteme SOLLTEN nach diesem Standard konfiguriert und einheitlich betrieben werden. Falls es bei einer Installation notwendig ist, vom Konfigurationsstandard abzuweichen, SOLLTEN alle Schritte von dem oder der ISB freigegeben und nachvollziehbar dokumentiert werden. Der Konfigurationsstandard SOLLTE regelmäßig überprüft und, falls erforderlich, angepasst werden.

APP.4.3.A13 Restriktive Handhabung von Datenbank-Links (S)

Es SOLLTE sichergestellt sein, dass nur Zuständige dazu berechtigt sind, Datenbank-Links (DB-Links) anzulegen. Werden solche Links angelegt, MÜSSEN so genannte Private DB-Links vor Public DB-Links bevorzugt angelegt werden. Alle von den Zuständigen angelegten DB-Links SOLLTEN dokumentiert und regelmäßig überprüft werden. Zudem SOLLTEN DB-Links mitberücksichtigt werden, wenn das Datenbanksystem gesichert wird (siehe APP.4.3.A9 *Datensicherung eines Datenbanksystems*).

APP.4.3.A14 ENTFALLEN (S)

Diese Anforderung ist entfallen.

APP.4.3.A15 ENTFALLEN (S)

Diese Anforderung ist entfallen.

APP.4.3.A16 Verschlüsselung der Datenbankanbindung (S)

Das Datenbankmanagementsystem SOLLTE so konfiguriert werden, dass Datenbankverbindungen immer verschlüsselt werden. Die dazu eingesetzten kryptografischen Verfahren und Protokolle SOLLTEN den internen Vorgaben der Institution entsprechen (siehe CON.1 *Kryptokonzept*).

APP.4.3.A17 Datenübernahme oder Migration (S) [Fachverantwortliche]

Es SOLLTE vorab definiert werden, wie initial oder regelmäßig Daten in eine Datenbank übernommen werden sollen. Nachdem Daten übernommen wurden, SOLLTE geprüft werden, ob sie vollständig und unverändert sind.

APP.4.3.A18 Überwachung des Datenbankmanagementsystems (S)

Die für den sicheren Betrieb kritischen Parameter, Ereignisse und Betriebszustände des Datenbankmanagementsystems SOLLTEN definiert werden. Diese SOLLTEN mithilfe eines Monitoring-Systems überwacht werden. Für alle kritischen Parameter, Ereignisse und Betriebszustände SOLLTEN Schwellwerte festgelegt werden. Wenn diese Werte überschritten werden, MUSS geeignet reagiert werden. Hierbei SOLLTEN die zuständigen Mitarbeitenden alarmiert werden. Anwendungsspezifische Parameter, Ereignisse, Betriebszustände und deren Schwellwerte SOLLTEN mit den Zuständigen für die Fachanwendungen abgestimmt werden.

APP.4.3.A19 Schutz vor schädlichen Datenbank-Skripten (S) [Entwickelnde]

Werden Datenbank-Skripte entwickelt, SOLLTEN dafür verpflichtende Qualitätskriterien definiert werden (siehe CON.8 *Software-Entwicklung*). Datenbank-Skripte SOLLTEN ausführlichen Funktionstests auf gesonderten Testsystemen unterzogen werden, bevor sie produktiv eingesetzt werden. Die Ergebnisse SOLLTEN dokumentiert werden.

APP.4.3.A20 Regelmäßige Audits (S)

Bei allen Komponenten des Datenbanksystems SOLLTE regelmäßig überprüft werden, ob alle festgelegten Sicherheitsmaßnahmen umgesetzt und diese korrekt konfiguriert sind. Dabei SOLLTE geprüft werden, ob der dokumentierte Stand dem Ist-Zustand entspricht und ob die Konfiguration des Datenbankmanagementsystems der dokumentierten Standardkonfiguration entspricht. Zudem SOLLTE geprüft werden, ob alle Datenbank-Skripte benötigt werden. Auch SOLLTE geprüft werden, ob sie dem Qualitätsstandard der Institution genügen. Zusätzlich SOLLTEN die Protokolldateien des Datenbanksystems und des Betriebssystems nach Auffälligkeiten untersucht werden (siehe DER.1 *Detektion von sicherheitsrelevanten Ereignissen*). Die Auditergebnisse SOLLTEN nachvollziehbar dokumentiert sein. Sie SOLLTEN mit dem Soll-Zustand abgeglichen werden. Abweichungen SOLLTE nachgegangen werden.

3.3. Anforderungen bei erhöhtem Schutzbedarf

Im Folgenden sind für diesen Baustein exemplarische Vorschläge für Anforderungen aufgeführt, die über dasjenige Schutzniveau hinausgehen, das dem Stand der Technik entspricht. Die Vorschläge SOLLTEN bei erhöhtem Schutzbedarf in Betracht gezogen werden. Die konkrete Festlegung erfolgt im Rahmen einer individuellen Risikoanalyse.

APP.4.3.A21 Einsatz von Datenbank Security Tools (H)

Es SOLLTEN Informationssicherheitsprodukte für Datenbanken eingesetzt werden. Die eingesetzten Produkte SOLLTEN folgende Funktionen bereitstellen:

- Erstellung einer Übersicht über alle Datenbanksysteme,
- erweiterte Konfigurationsmöglichkeiten und Rechtemanagement der Datenbanken,
- Erkennung und Unterbindung von möglichen Angriffen (z. B. Brute Force Angriffe auf Konten, SQL-Injection) und
- Auditfunktionen (z. B. Überprüfung von Konfigurationsvorgaben).

APP.4.3.A22 Notfallvorsorge (H)

Für das Datenbankmanagementsystem SOLLTE ein Notfallplan erstellt werden, der festlegt, wie ein Notbetrieb realisiert werden kann. Die für den Notfallplan notwendigen Ressourcen SOLLTEN ermittelt werden. Zusätzlich SOLLTE der Notfallplan definieren, wie aus dem Notbetrieb der Regelbetrieb wiederhergestellt werden kann. Der Notfallplan SOLLTE die nötigen Meldewege, Reaktionswege, Ressourcen und Reaktionszeiten der Fachverantwortlichen festlegen. Auf Basis eines Koordinationsplans zum Wiederanlauf SOLLTEN alle von der Datenbank abhängigen IT-Systeme vorab ermittelt und berücksichtigt werden.

APP.4.3.A23 Archivierung (H)

Ist es erforderlich, Daten eines Datenbanksystems zu archivieren, SOLLTE ein entsprechendes Archivierungskonzept erstellt werden. Es SOLLTE sichergestellt sein, dass die Datenbestände zu einem späteren Zeitpunkt wieder vollständig und konsistent verfügbar sind.

Im Archivierungskonzept SOLLTEN sowohl die Intervalle der Archivierung als auch die Vorhaltefristen der archivierten Daten festgelegt werden. Zusätzlich SOLLTE dokumentiert werden, mit welcher Technik die Datenbanken archiviert wurden. Mit den archivierten Daten SOLLTEN regelmäßig Wiederherstellungstests durchgeführt werden. Die Ergebnisse SOLLTEN dokumentiert werden.

APP.4.3.A24 Datenverschlüsselung in der Datenbank (H)

Die Daten in den Datenbanken SOLLTEN verschlüsselt werden. Dabei SOLLTEN vorher unter anderem folgende Faktoren betrachtet werden:

- Einfluss auf die Performance,
- Schlüsselverwaltungsprozesse und -verfahren, einschließlich separater Schlüsselaufbewahrung und -sicherung,
- Einfluss auf Backup-Recovery-Konzepte,
- funktionale Auswirkungen auf die Datenbank, beispielsweise Sortiermöglichkeiten.

APP.4.3.A25 Sicherheitsprüfungen von Datenbanksystemen (H)

Datenbanksysteme SOLLTEN regelmäßig mithilfe von Sicherheitsprüfungen kontrolliert werden. Bei den Sicherheitsprüfungen SOLLTEN die systemischen und spezifischen Aspekte des herstellenden Unternehmens der eingesetzten Datenbank-Infrastruktur (z. B. Verzeichnisdienste) sowie des eingesetzten Datenbankmanagementsystems betrachtet werden.

4. Weiterführende Informationen

4.1. Wissenswertes

Das BSI hat im Rahmen der Allianz für Cybersicherheit zum Themenfeld Datenbanksicherheit folgende Partnerbeiträge veröffentlicht:

- Oracle: Datenbank-Sicherheit – Grundüberlegungen
- McAfee: Datenbanksicherheit in Virtualisierungs- und Cloud-Computing-Umgebungen

Die Deutsche Telekom Gruppe hat im Rahmen ihres Privacy and Security Assessment Verfahrens (PSA-Verfahren) das Dokument „Sicherheitsanforderung Datenbanksysteme“ zum Themenfeld Datenbanken veröffentlicht.

Das Information Security Forum (ISF) macht in seinem Standard „The Standard of Good Practice for Information Security“ im Kapitel BA2.3 Protection of Databases Vorgaben für die Absicherung von relationalen Datenbanksystemen.



APP.4.4 Kubernetes

1. Beschreibung

1.1. Einleitung

Kubernetes hat sich als De-Facto-Standard für die Orchestrierung von Containern in der Public und Private Cloud etabliert. Auch für IoT und andere Anwendungsfälle ist Kubernetes im Einsatz, mit K3S gibt es beispielsweise eine Edition, die für sehr kleine Server wie Einplatinencomputer gedacht ist. Auch der sogenannte Cloud Native Stack, der aus vielen verschiedenen Komponenten besteht, baut auf dem von Kubernetes etablierten Standard auf.

Der Begriff *Container* bezeichnet eine Technik, bei der ein Host-System Anwendungen parallel in separierten Umgebungen ausführt (Operating System Level Virtualization). In den meisten Fällen erfolgt die Überwachung, das Starten und Beenden und die weitere Verwaltung der Container durch eine Verwaltungssoftware, die somit die sogenannte *Orchestrierung* übernimmt. Kubernetes fasst dabei einen oder mehrere zusammengehörige Container in einem *Pod* zusammen. Da der Fokus des Bausteins auf Kubernetes liegt, wird im Weiteren nur von Pods und nicht von einzelnen Containern gesprochen. Die Orchestrierung erfolgt dabei zumeist in Gruppen von gemeinsam verwalteten Kubernetes-Nodes in einem oder mehreren sogenannten *Clustern*.

Um die Orchestrierung von Pods zu betreiben und diese zu verwalten, haben sich mehrere Produkte etabliert, die es erlauben, auch sehr große Umgebungen zu bedienen. In ihrem Kern setzen allerdings alle auf Kubernetes auf. Bei der Betrachtung ist zwischen der *Runtime*, die die Prozesse auf den Kubernetes-Nodes betreibt, und der *Orchestrierung*, die die Runtimes auf mehreren Kubernetes-Nodes steuert, zu unterscheiden.

Zusätzlich zu diesen beiden zentralen Komponenten besteht der Betrieb von Kubernetes in den meisten Fällen noch aus einer spezialisierten Infrastruktur, zu der z. B. Registries, Code-Versionierung und -Speicherung, Automatisierungswerzeuge, Verwaltungsserver, Speichersysteme oder virtuelle Netze gehören.

Die folgenden Begriffe werden im Baustein in dieser Bedeutung verwendet:

- *Anwendung* bezeichnet eine Zusammenstellung mehrerer Programme, die gemeinsam eine Aufgabe erfüllen
- *Cluster* sind Betriebsumgebungen für Container mit mehreren Nodes
- *Container* sind aus einem Image gestartete Prozesse, die innerhalb von Betriebssystem-Namespaces laufen
- *Container Network Interface (CNI)* bezeichnet die Schnittstelle zur Verwaltung der virtuellen Netze im Cluster
- *Container Storage Interface (CSI)* bezeichnet die Schnittstelle zu den zumeist externen Speichersystemen, die Kubernetes den Pods bereitstellen kann
- *Control Plane* bezeichnet alle für die Verwaltung, also Orchestrierung der Nodes, Runtimes und Cluster eingesetzten Anwendungen
- *Images* sind alle der Open Container Initiative (OCI) entsprechenden Software-Pakete, diese umfassen sowohl Basis-Images für eigene Images als auch solche Images, die unverändert im Einsatz sind
- *Node* bezeichnet einen Server, der für den Betrieb der Runtime installiert und optimiert ist
- *Pod* bezeichnet eine Sammlung mehrerer Container, die innerhalb der gleichen Betriebssystem-Namespaces laufen
- *Registry* ist der Oberbegriff für die Code-Verwaltung und die Speicherung der Images
- *Runtime* bezeichnet die Software, die die Software im Image als Container startet

1.2. Zielsetzung

Ziel dieses Bausteins ist der Schutz von Informationen, die in Kubernetes-Clustern verarbeitet, angeboten oder darüber übertragen werden.

1.3. Abgrenzung und Modellierung

Der Baustein APP.4.4 *Kubernetes* ist immer zusammen mit dem Baustein SYS.1.6 *Containerisierung* anzuwenden. Dabei ist es bezogen auf den Fokus des vorliegenden Bausteins nicht relevant, welche Container-Runtime im Einsatz ist oder welche zusätzlichen Anwendungen Teil der Control Plane sind.

Der Baustein enthält grundsätzliche Anforderungen zur Einrichtung, zum Betrieb und zur Orchestrierung mit Kubernetes sowie zur spezialisierten Infrastruktur, die zum Betrieb notwendig ist. Letzteres beinhaltet Registries, CSI/CNI, Nodes und Automatisierungssoftware, soweit sie direkt mit dem Cluster interagieren. Die Anforderungen für diese Anwendungen beziehen sich zumeist auf die Schnittstellen, enthalten aber auch Anforderungen, die den Betrieb dieser Anwendungen betreffen, sofern sie direkt die Sicherheit des Clusters berühren. Weitere im Kubernetes-Umfeld übliche Dienste, wie z. B. Automatisierung für CI/CD-Pipelines und Codeverwaltung in z. B. Git, behandelt der Baustein nicht in der Tiefe.

Der Baustein modelliert umfassend einen Cluster. Die Anwendungen der Control Plane, Dienste zur Automatisierung und die Nodes, sind hier als eine Gruppe zu sehen und zu behandeln.

Sicherheitsanforderungen für die in Kubernetes-Clustern betriebenen Dienste, wie z. B. Webserver (APP.3.2 *Webserver*) oder E-Mail-Server (siehe APP.5.3 *Allgemeiner E-Mail-Client und -Server*), sind Gegenstand eigener Bausteine.

2. Gefährdungslage

Da IT-Grundschutz-Bausteine nicht auf individuelle Informationsverbünde eingehen können, werden zur Darstellung der Gefährdungslage typische Szenarien zugrunde gelegt. Die folgenden spezifischen Bedrohungen und Schwachstellen sind für den Baustein APP.4.4 *Kubernetes* von besonderer Bedeutung.

2.1. Mangelhafte Authentisierung und Autorisierung in der Control Plane

Um Runtimes, Nodes und auch Kubernetes selbst zu verwalten, benötigen sowohl Administrierende als auch die toolgestützte Bereitstellung administrative Zugänge. Diese Zugänge sind entweder als Unix-Sockets oder Netzports ausgeführt. Mechanismen zur Authentisierung und Verschlüsselung der administrativen Zugänge sind häufig vorhanden, aber nicht bei allen Produkten standardmäßig aktiviert.

Wenn Unbefugte auf das Datennetz oder auf die Nodes zugreifen, können sie über ungeschützte administrative Zugänge Befehle ausführen, die der Verfügbarkeit, Vertraulichkeit und Integrität der verarbeiteten Daten schaden können.

2.2. Vertraulichkeitsverlust von Zugangsdaten

Oft benötigen Pods Zugangsdaten (Access Token) für Kubernetes. Über einen Angriff auf den Pod können diese Zugangsdaten in unbefugte Hände gelangen. Mit diesen Zugangsdaten ist es bei Angriffen möglich, mit der Control Plane authentifiziert zu interagieren und, sofern die Berechtigungen ausreichen, auch Veränderungen an der Orchestrierung vorzunehmen.

2.3. Ressourcenkonflikte auf Nodes

Einzelne Pods können den Node oder auch die Orchestrierung überlasten und so die Verfügbarkeit aller anderen Pods auf dem Node oder auch den Betrieb des Nodes selbst gefährden.

2.4. Unautorisierte Änderungen an Clustern

Die Automatisierung mit CI/CD und die daraus folgende Notwendigkeit, den Werkzeugen privilegierte Zugangsberichtigungen zu erteilen, birgt das Risiko, dass nicht autorisierte Änderungen an Clustern erfolgen. So kann z. B. eine neue Version einer Anwendung auf dem Cluster aufgebracht werden, die nicht ausreichend getestet ist oder

die den Freigabeprozess nicht durchlaufen hat. Auch ist es bei Fehlern in den Berechtigungen auf der CI/CD-Umgebung möglich, dass Schadsoftware in die Cluster eindringen und dort Daten auslesen, löschen oder verändern kann.

2.5. Unberechtigte Kommunikation

Alle Pods in einem Cluster sind grundsätzlich in der Lage, miteinander, mit den Nodes im eigenen Cluster sowie beliebigen anderen IT-Systemen zu kommunizieren. Sofern diese Kommunikation nicht eingeschränkt ist, kann dies ausgenutzt werden, um z. B. die Control Plane, andere Pods oder die Nodes anzugreifen.

Auch besteht die Gefahr, dass Pods im Cluster unerwünscht von außen erreichbar sind. So kann ein Angriff gegen Dienste, die eigentlich nur innerhalb des Clusters erreichbar sein sollten, von außen erfolgen. Diese Gefährdung wird durch die geringere Aufmerksamkeit verschlimmert, die internen Diensten oft entgegengebracht wird. Wird z. B. eine Verwundbarkeit auf einem nur intern eingesetzten Dienst toleriert und ist dieser auch von außen erreichbar, gefährdet dies den gesamten Cluster erheblich.

3. Anforderungen

Im Folgenden sind die spezifischen Anforderungen des Bausteins APP.4.4 *Kubernetes* aufgeführt. Der oder die Informationssicherheitsbeauftragte (ISB) ist dafür zuständig, dass alle Anforderungen gemäß dem festgelegten Sicherheitskonzept erfüllt und überprüft werden. Bei strategischen Entscheidungen ist der oder die ISB stets einzubeziehen.

Im IT-Grundschutz-Kompendium sind darüber hinaus weitere Rollen definiert. Sie sollten besetzt werden, insofern dies sinnvoll und angemessen ist.

Zuständigkeiten	Rollen
Grundsätzlich zuständig	IT-Betrieb
Weitere Zuständigkeiten	Keine

Genau eine Rolle sollte *Grundsätzlich zuständig* sein. Darüber hinaus kann es noch *Weitere Zuständigkeiten* geben. Falls eine dieser weiteren Rollen für die Erfüllung einer Anforderung vorrangig zuständig ist, dann wird diese Rolle hinter der Überschrift der Anforderung in eckigen Klammern aufgeführt. Die Verwendung des Singulärs oder Plurals sagt nichts darüber aus, wie viele Personen diese Rollen ausfüllen sollen.

3.1. Basis-Anforderungen

Die folgenden Anforderungen MÜSSEN für diesen Baustein vorrangig erfüllt werden.

APP.4.4.A1 Planung der Separierung der Anwendungen (B)

Vor der Inbetriebnahme MUSS geplant werden, wie die in den Pods betriebenen Anwendungen und deren unterschiedlichen Test- und Produktions-Betriebsumgebungen separiert werden. Auf Basis des Schutzbedarfs der Anwendungen MUSS festgelegt werden, welche Architektur der Namespaces, Meta-Tags, Cluster und Netze angepasst auf die Risiken eingeht und ob auch virtualisierte Server und Netze zum Einsatz kommen sollen.

Die Planung MUSS Regelungen zu Netz-, CPU- und Festspeicherseparierung enthalten. Die Separierung SOLLTE auch das Netzzonenkonzept und den Schutzbedarf beachten und auf diese abgestimmt sein.

Anwendungen SOLLTEN jeweils in einem eigenen Kubernetes-Namespace laufen, der alle Programme der Anwendung umfasst. Nur Anwendungen mit ähnlichem Schutzbedarf und ähnlichen möglichen Angriffsvektoren SOLLTEN einen Kubernetes-Cluster teilen.

APP.4.4.A2 Planung der Automatisierung mit CI/CD (B)

Wenn eine Automatisierung des Betriebs von Anwendungen in Kubernetes mithilfe von CI/CD stattfindet, DARF diese NUR nach einer geeigneten Planung erfolgen. Die Planung MUSS den gesamten Lebenszyklus von Inbetriebnis bis Außerbetriebnahme inklusive Entwicklung, Tests, Betrieb, Überwachung und Updates umfassen. Das Rollen- und Rechtekonzept sowie die Absicherung von Kubernetes Secrets MÜSSEN Teil der Planung sein.

APP.4.4.A3 Identitäts- und Berechtigungsmanagement bei Kubernetes (B)

Kubernetes und alle anderen Anwendungen der Control Plane MÜSSEN jede Aktion eines Benutzenden oder, im automatisierten Betrieb, einer entsprechenden Software authentifizieren und autorisieren, unabhängig davon, ob die Aktionen über einen Client, eine Weboberfläche oder über eine entsprechende Schnittstelle (API) erfolgt. Administrative Handlungen DÜRFEN NICHT anonym erfolgen.

Benutzende DÜRFEN NUR die unbedingt notwendigen Rechte erhalten. Berechtigungen ohne Einschränkungen MÜSSEN sehr restriktiv vergeben werden.

Nur ein kleiner Kreis von Personen SOLLTE berechtigt sein, Prozesse der Automatisierung zu definieren. Nur ausgewählte Administrierende SOLLTEN in Kubernetes das Recht erhalten, Freigaben für Festspeicher (Persistent Volumes) anzulegen oder zu ändern.

APP.4.4.A4 Separierung von Pods (B)

Der Betriebssystem-Kernel der Nodes MUSS über Isolationsmechanismen zur Beschränkung von Sichtbarkeit und Ressourcennutzung der Pods untereinander verfügen (vergleiche Linux Namespaces und cgroups). Die Trennung MUSS dabei mindestens IDs von Prozessen sowie Benutzenden, Inter-Prozess-Kommunikation, Dateisystem und Netz inklusive Hostname umfassen.

APP.4.4.A5 Datensicherung im Cluster (B)

Es MUSS eine Datensicherung des Clusters erfolgen. Die Datensicherung MUSS umfassen:

- Festspeicher (Persistent Volumes),
- Konfigurationsdateien von Kubernetes und den weiteren Programmen der Control Plane,
- den aktuellen Zustand des Kubernetes-Clusters inklusive der Erweiterungen,
- Datenbanken der Konfiguration, namentlich hier *etcd*,
- alle Infrastrukturanwendungen, die zum Betrieb des Clusters und der darin befindlichen Dienste notwendig sind und
- die Datenhaltung der Code und Image Registries.

Es SOLLTEN auch Snapshots für den Betrieb der Anwendungen betrachtet werden. Snapshots DÜRFEN die Datensicherung NICHT ersetzen.

3.2. Standard-Anforderungen

Gemeinsam mit den Basis-Anforderungen entsprechen die folgenden Anforderungen dem Stand der Technik für diesen Baustein. Sie SOLLTEN grundsätzlich erfüllt werden.

APP.4.4.A6 Initialisierung von Pods (S)

Sofern im Pod zum Start eine Initialisierung z. B. einer Anwendung erfolgt, SOLLTE diese in einem eigenen Init-Container stattfinden. Es SOLITE sichergestellt sein, dass die Initialisierung alle bereits laufenden Prozesse beendet. Kubernetes SOLLTE nur bei erfolgreicher Initialisierung die weiteren Container starten.

APP.4.4.A7 Separierung der Netze bei Kubernetes (S)

Die Netze für die Administration der Nodes, der Control Plane sowie die einzelnen Netze der Anwendungsdienste SOLLTEN separiert werden.

Es SOLLTEN nur die für den Betrieb notwendigen Netzports der Pods in die dafür vorgesehenen Netze freigegeben werden. Bei mehreren Anwendungen auf einem Kubernetes-Cluster SOLLTEN zunächst alle Netzverbindungen zwischen den Kubernetes-Namespace untersagt und nur benötigte Netzverbindungen gestattet sein (Whitelisting). Die zur Administration der Nodes, der Runtime und von Kubernetes inklusive seiner Erweiterungen notwendigen Netzports SOLLTEN nur aus dem Administrationsnetz und von Pods, die diese benötigen, erreichbar sein.

Nur ausgewählte Administrierende SOLLTEN in Kubernetes berechtigt sein, das CNI zu verwalten und Regeln für das Netz anzulegen oder zu ändern.

APP.4.4.A8 Absicherung von Konfigurationsdateien bei Kubernetes (S)

Die Konfigurationsdateien des Kubernetes-Clusters, inklusive aller Erweiterungen und Anwendungen, SOLLTEN versioniert und annotiert werden.

Zugangsrechte auf die Verwaltungssoftware der Konfigurationsdateien SOLLTEN minimal vergeben werden. Zugriffsrechte für lesenden und schreibenden Zugriff auf die Konfigurationsdateien der Control Plane SOLLTEN besonders sorgfältig vergeben und eingeschränkt sein.

APP.4.4.A9 Nutzung von Kubernetes Service-Accounts (S)

Pods SOLLTEN NICHT den „default“-Service-Account nutzen. Dem „default“-Service-Account SOLLTEN keine Rechte eingeräumt werden. Pods für unterschiedliche Anwendungen SOLLTEN jeweils unter eigenen Service-Accounts laufen. Berechtigungen für die Service-Accounts der Pods der Anwendungen SOLLTEN auf die unbedingt notwendigen Rechte beschränkt werden.

Pods, die keinen Service-Account benötigen, SOLLTEN diesen nicht einsehen können und keinen Zugriff auf entsprechende Token haben.

Nur Pods der Control Plane und Pods, die diese unbedingt benötigen, SOLLTEN privilegierte Service-Accounts nutzen.

Programme der Automatisierung SOLLTEN jeweils eigene Token erhalten, auch wenn sie aufgrund ähnlicher Aufgaben einen gemeinsamen Service-Account nutzen.

APP.4.4.A10 Absicherung von Prozessen der Automatisierung (S)

Alle Prozesse der Automatisierungssoftware, wie CI/CD und deren Pipelines, SOLLTEN nur mit unbedingt notwendigen Rechten arbeiten. Wenn unterschiedliche Gruppen von Benutzenden die Konfiguration über die Automatisierungssoftware verändern oder Pods starten können, SOLLTE dies für jede Gruppe durch eigene Prozesse durchgeführt werden, die nur die für die jeweilige Gruppe notwendigen Rechte besitzen.

APP.4.4.A11 Überwachung der Container (S)

In Pods SOLLTE jeder Container einen Health Check für den Start und den Betrieb („readiness“ und „liveness“) definieren. Diese Checks SOLLTEN Auskunft über die Verfügbarkeit der im Pod ausgeführten Software geben. Die Checks SOLLTEN fehlschlagen, wenn die überwachte Software ihre Aufgaben nicht ordnungsgemäß wahrnehmen kann. Für jede dieser Kontrollen SOLLTE eine dem im Pod betriebenen Dienst angemessene Zeitspanne definieren. Auf Basis dieser Checks SOLLTE Kubernetes die Pods löschen oder neu starten.

APP.4.4.A12 Absicherung der Infrastruktur-Anwendungen (S)

Sofern eine eigene Registry für Images oder eine Software zur Automatisierung, zur Verwaltung des Festspeichers, zur Speicherung von Konfigurationsdateien oder ähnliches im Einsatz ist, SOLLTE deren Absicherung mindestens betrachten:

- Verwendung von personenbezogenen und Service-Accounts für den Zugang,
- verschlüsselte Kommunikation auf allen Netzports,
- minimale Vergabe der Berechtigungen an Benutzende und Service Accounts,
- Protokollierung der Veränderungen und
- regelmäßige Datensicherung.

3.3. Anforderungen bei erhöhtem Schutzbedarf

Im Folgenden sind für diesen Baustein exemplarische Vorschläge für Anforderungen aufgeführt, die über dasjenige Schutzniveau hinausgehen, das dem Stand der Technik entspricht. Die Vorschläge SOLLTEN bei erhöhtem Schutzbedarf in Betracht gezogen werden. Die konkrete Festlegung erfolgt im Rahmen einer individuellen Risikoanalyse.

APP.4.4.A13 Automatisierte Auditierung der Konfiguration (H)

Es SOLLTE ein automatisches Audit der Einstellungen der Nodes, von Kubernetes und der Pods der Anwendungen gegen eine definierte Liste der erlaubten Einstellungen und gegen standardisierte Benchmarks erfolgen.

Kubernetes SOLLTE die aufgestellten Regeln im Cluster durch Anbindung geeigneter Werkzeuge durchsetzen.

APP.4.4.A14 Verwendung dedizierter Nodes (H)

In einem Kubernetes-Cluster SOLLTEN die Nodes dedizierte Aufgaben zugewiesen bekommen und jeweils nur Pods betreiben, welche der jeweiligen Aufgabe zugeordnet sind.

Bastion Nodes SOLLTEN alle ein- und ausgehenden Datenverbindungen der Anwendungen zu anderen Netzen übernehmen.

Management Nodes SOLLTEN die Pods der Control Plane betreiben und sie SOLLTEN nur die Datenverbindungen der Control Plane übernehmen.

Sofern eingesetzt, SOLLTEN Speicher-Nodes nur die Pods der Festspeicherdienste im Cluster betreiben.

APP.4.4.A15 Trennung von Anwendungen auf Node- und Cluster-Ebene (H)

Anwendungen mit einem sehr hohen Schutzbedarf SOLLTEN jeweils eigene Kubernetes-Cluster oder dedizierte Nodes nutzen, die nicht für andere Anwendungen bereitstehen.

APP.4.4.A16 Verwendung von Operatoren (H)

Die Automatisierung von Betriebsaufgaben in Operatoren SOLLTE bei besonders kritischen Anwendungen und den Programmen der Control Plane zum Einsatz kommen.

APP.4.4.A17 Attestierung von Nodes (H)

Nodes SOLLTEN eine kryptografisch und möglichst mit einem TPM verifizierte gesicherte Zustandsmeldung an die Control Plane senden. Die Control Plane SOLLTE nur Nodes in den Cluster aufnehmen, die erfolgreich ihre Unverzerrtheit nachweisen konnten.

APP.4.4.A18 Verwendung von Mikro-Segmentierung (H)

Die Pods SOLLTEN auch innerhalb eines Kubernetes-Namespace nur über die notwendigen Netzports miteinander kommunizieren können. Es SOLLTEN Regeln innerhalb des CNI existieren, die alle bis auf die für den Betrieb notwendigen Netzverbindungen innerhalb des Kubernetes-Namespace unterbinden. Diese Regeln SOLLTEN Quelle und Ziel der Verbindungen genau definieren und dafür mindestens eines der folgenden Kriterien nutzen: Service-Name, Metadaten („Labels“), die Kubernetes Service Accounts oder zertifikatsbasierte Authentifizierung.

Alle Kriterien, die als Bezeichnung für diese Verbindung dienen, SOLLTEN so abgesichert sein, dass sie nur von berechtigten Personen und Verwaltungs-Diensten verändert werden können.

APP.4.4.A19 Hochverfügbarkeit von Kubernetes (H)

Der Betrieb SOLLTE so aufgebaut sein, dass bei Ausfall eines Standortes die Cluster und damit die Anwendungen in den Pods entweder ohne Unterbrechung weiterlaufen oder in kurzer Zeit an einem anderen Standort neu anlaufen können.

Für den Wiederanlauf SOLLTEN alle notwendigen Konfigurationsdateien, Images, Nutzdaten, Netzverbindungen und sonstige für den Betrieb benötigten Ressourcen inklusive der zum Betrieb nötigen Hardware bereits an diesem Standort verfügbar sein.

Für den unterbrechungsfreien Betrieb des Clusters SOLLTEN die Control Plane von Kubernetes, die Infrastruktur-Anwendungen der Cluster sowie die Pods der Anwendungen anhand von Standort-Daten der Nodes über mehrere Brandabschnitte so verteilt werden, dass der Ausfall eines Brandabschnitts nicht zum Ausfall der Anwendung führt.

APP.4.4.A20 Verschlüsselte Datenhaltung bei Pods (H)

Die Dateisysteme mit den persistenten Daten der Control Plane (hier besonders `etcd`) und der Anwendungsdienste SOLLTEN verschlüsselt werden.

APP.4.4.A21 Regelmäßiger Restart von Pods (H)

Bei einem erhöhten Risiko für Fremdeinwirkung und einem sehr hohen Schutzbedarf SOLLTEN Pods regelmäßig gestoppt und neu gestartet werden. Kein Pod SOLLTE länger als 24 Stunden laufen. Dabei SOLLTE die Verfügbarkeit der Anwendungen im Pod sichergestellt sein.

4. Weiterführende Informationen

4.1. Wissenswertes

Weiterführende Informationen zu Gefährdungen und Sicherheitsmaßnahmen im Bereich Container finden sich unter anderem in folgenden Veröffentlichungen:

- NIST 800-190
<https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-190.pdf>
- CIS Benchmark Kubernetes
<https://www.cisecurity.org/benchmark/kubernetes/>
- OCI – Open Container Initiative
<https://www.opencontainers.org/>
- CNCF – Cloud Native Computing Foundation
<https://www.cncf.io/>



APP.4.6 SAP ABAP-Programmierung

1. Beschreibung

1.1. Einleitung

Häufig werden in SAP-Systemen Eigenentwicklungen programmiert. Die Gründe dafür sind vielfältig, so können Geschäftsprozesse oder Anforderungen an das Reporting mit Hilfe von Eigenentwicklungen individuell an die Institution angepasst werden. Außerdem ist es möglich, spezielle Funktionen zu erstellen, die in der Standard-Auslieferung nicht vorhanden sind.

Eigenentwicklungen werden von Entwickelnden der Institution oder von beauftragten Entwickelnden programmiert. Im SAP-Umfeld wird dazu häufig ABAP (Advanced Business Application Programming) verwendet.

ABAP ist eine proprietäre, plattformunabhängige Programmiersprache des Unternehmens SAP. Sie wurde für die Programmierung kommerzieller Anwendungen im SAP-Umfeld entwickelt und ähnelt in ihrer Grundstruktur entfernt der Sprache COBOL. Wichtige Merkmale sind:

- Integration eines Authentisierungs-, Rollen- und Berechtigungskonzepts,
- Verwendung eines proprietären, datenbankunabhängigen SQL-Derivats (Open SQL),
- Unterstützung der Kommunikation zwischen verschiedenen SAP-Systemen sowie
- Integration von Audit-Optionen.

1.2. Zielsetzung

Der Baustein zeigt ABAP-Entwickelnden und Sicherheitstestenden relevante technische Risiken auf, die sich durch ABAP-Eigenentwicklungen ergeben können. Außerdem werden Anforderungen definiert, die aufzeigen, wie ABAP-Programme sicher entwickelt und eingesetzt werden können.

Der Baustein setzt grundlegende Kenntnisse in ABAP und im Umgang mit ABAP-Entwicklungswerkzeugen voraus.

1.3. Abgrenzung und Modellierung

Der Baustein APP.4.6 SAP ABAP-Programmierung ist auf jedes SAP-System einmal anzuwenden, wenn Eigenentwicklungen in der Programmiersprache ABAP erstellt werden.

Mit diesem Baustein werden die Bausteine CON.8 Software-Entwicklung, APP.6 Allgemeine Software und APP.7 Entwicklung von Individualsoftware um konkrete Aspekte zur Entwicklung von ABAP-Programmen erweitert.

Der Baustein stellt keine vollständige Anleitung dar, um ABAP-Programme zu entwickeln, sondern beschreibt die generellen Risiken der Programmiersprache ABAP. Im Baustein werden Anforderungen definiert, die bei der Entwicklung von ABAP-Programmen aus Sicherheitssicht erfüllt werden sollten.

Da Webanwendungen nur einen sehr geringen Anteil aller ABAP-Anwendungen in SAP-Implementierungen ausmachen, stehen Web-Schwachstellen nicht im Fokus dieses Dokuments.

2. Gefährdungslage

Da IT-Grundschutz-Bausteine nicht auf individuelle Informationsverbünde eingehen können, werden zur Darstellung der Gefährdungslage typische Szenarien zugrunde gelegt. Die folgenden spezifischen Bedrohungen und Schwachstellen sind für den Baustein APP.4.6 SAP ABAP-Programmierung von besonderer Bedeutung.

2.1. Fehlende Berechtigungsprüfungen

In SAP werden Berechtigungen nur dann geprüft, wenn eine entsprechende Berechtigungsprüfung von Entwickelnden im Programm implementiert wurde. Ohne eine solche Prüfung im Programm-Code wird also nicht getestet, ob Benutzende auch wirklich berechtigt sind eine Aktion auszuführen. In selbst entwickeltem Programm-Code werden Berechtigungsprüfungen aber häufig vergessen. Somit greift das gesamte Berechtigungskonzept oftmals nicht und unberechtigte Personen können auf die im SAP-System gespeicherten Daten zugreifen. Dadurch kann etwa auch gegen Compliance-Anforderungen verstoßen werden. Dies kann besonders bei Wirtschaftsprüfungen schwerwiegende Folgen nach sich ziehen.

2.2. Verlust von Vertraulichkeit oder Integrität von kritischen Daten

SAP-Systeme enthalten viele institutionskritische Informationen. Der SAP-Standard sieht verschiedene Mechanismen vor, diese Daten zu schützen. Allerdings könnte durch fehlerhafte ABAP-Eigenentwicklungen unerlaubt auf institutionskritische Informationen zugegriffen werden. Mitarbeitende oder Angreifende könnten die Daten so in eine nicht mehr kontrollierbare Umgebung transferieren. Ebenso könnten mit Hilfe von ABAP-Programmen kritische Daten manipuliert werden, indem die Sicherheitsmechanismen des SAP-Standards umgangen werden.

2.3. Injection-Schwachstellen

Injection-Schwachstellen entstehen dadurch, dass Angreifende Steuerzeichen bzw. Kommandos über das Eingabefeld in eine Anwendung einschleust. Ein erfolgreicher Angriff kann den geplanten Programmablauf durch unerwartete Kommandos stören.

Injection-Schwachstellen stellen für Eigenentwicklungen das größte Sicherheitsrisiko dar. Durch fehlerhaften Code in einer ABAP-Anwendung können Angreifende ein SAP-System mitunter vollständig kontrollieren. Da solche Angriffe sehr komplex sind und es viele Varianten davon gibt, lassen sie sich ohne spezielle Schulungen kaum erkennen und beheben.

2.4. Umgehung von vorhandenen SAP-Sicherheitsmechanismen

Der SAP-Standard stellt verschiedene Schutzmechanismen für Daten zur Verfügung. Dazu gehören unter anderem die Mandantentrennung, Identity-Management sowie Rollen und Berechtigungen. Diese Sicherheitsmechanismen können im Code jedoch bewusst umgangen oder ungewollt weggelassen werden.

3. Anforderungen

Im Folgenden sind die spezifischen Anforderungen des Bausteins APP.4.6 *SAP ABAP-Programmierung* aufgeführt. Der oder die Informationssicherheitsbeauftragte (ISB) ist dafür zuständig, dass alle Anforderungen gemäß dem festgelegten Sicherheitskonzept erfüllt und überprüft werden. Bei strategischen Entscheidungen ist der oder die ISB stets einzubeziehen.

Im IT-Grundschutz-Kompendium sind darüber hinaus weitere Rollen definiert. Sie sollten besetzt werden, insofern dies sinnvoll und angemessen ist.

Zuständigkeiten	Rollen
Grundsätzlich zuständig	Entwickelnde
Weitere Zuständigkeiten	Keine

Genau eine Rolle sollte *Grundsätzlich zuständig* sein. Darüber hinaus kann es noch *Weitere Zuständigkeiten* geben. Falls eine dieser weiteren Rollen für die Erfüllung einer Anforderung vorrangig zuständig ist, dann wird diese Rolle hinter der Überschrift der Anforderung in eckigen Klammern aufgeführt. Die Verwendung des Singulars oder Plurals sagt nichts darüber aus, wie viele Personen diese Rollen ausfüllen sollen.

3.1. Basis-Anforderungen

Die folgenden Anforderungen MÜSSEN für diesen Baustein vorrangig erfüllt werden:

APP.4.6.A1 Absicherung von Reports mit Berechtigungsprüfungen (B)

Es MUSS sichergestellt sein, dass nur berechtigte Personen selbst programmierte Auswertungen (Reports) starten können. Deswegen MUSS jeder Report explizite, zum Kontext passende Berechtigungsprüfungen durchführen.

APP.4.6.A2 Formal korrekte Auswertung von Berechtigungsprüfungen (B)

Jede Berechtigungsprüfung im Code MUSS durch Abfrage des Rückgabewertes *SY-SUBRC* ausgewertet werden.

APP.4.6.A3 Berechtigungsprüfung vor dem Start einer Transaktion (B)

Wenn Entwickelnde den Befehl *CALL TRANSACTION* verwenden, MUSS vorher immer eine Startberechtigungsprüfung durchgeführt werden.

APP.4.6.A4 Verzicht auf proprietäre Berechtigungsprüfungen (B)

Jede Berechtigungsprüfung MUSS technisch über den dafür vorgesehenen Befehl *AUTHORITY-CHECK* erfolgen. Proprietäre Berechtigungsprüfungen, z. B. basierend auf Konto-Kennungen, DÜRFEN NICHT benutzt werden.

3.2. Standard-Anforderungen

Gemeinsam mit den Basis-Anforderungen entsprechen die folgenden Anforderungen dem Stand der Technik für diesen Baustein. Sie SOLLTEN grundsätzlich erfüllt werden.

APP.4.6.A5 Erstellung einer Richtlinie für die ABAP-Entwicklung (S)

Es SOLLTE eine Richtlinie für die Entwicklung von ABAP-Programmen erstellt werden. Die Richtlinie SOLLTE neben Namenskonventionen auch Vorgaben zu ABAP-Elementen beinhalten, die verwendet bzw. nicht verwendet werden dürfen. Die Anforderungen aus diesem Baustein SOLLTEN in die Richtlinie aufgenommen werden. Die Richtlinie SOLLTE für die Entwickelnden verbindlich sein.

APP.4.6.A6 Vollständige Ausführung von Berechtigungsprüfungen (S)

Bei einer Berechtigungsprüfung im ABAP-Code (*AUTHORITY-CHECK <OBJECT>*) SOLLTE sichergestellt sein, dass alle Felder des relevanten Berechtigungsobjekts überprüft werden. Wenn einzelne Felder tatsächlich nicht benötigt werden, SOLLTEN sie als *DUMMY* gekennzeichnet werden. Zusätzlich SOLLTE am Feld der Grund für die Ausnahme dokumentiert werden.

APP.4.6.A7 Berechtigungsprüfung während der Eingabeverarbeitung (S)

Funktionscodes und Bildschirmelemente von ABAP-Dynpro-Anwendungen SOLLTEN konsistent sein. Wenn ein Bildschirmelement abgeschaltet wurde, dann SOLLTE eine Anwendung NICHT ohne adäquate Berechtigungsprüfungen auf Ereignisse dieses Elements reagieren. Wenn bestimmte Einträge eines Dynpro-Menüs ausgeblendet oder einzelne Schaltflächen deaktiviert werden, dann SOLLTEN auch die zugehörigen Funktionscodes nicht ausgeführt werden.

APP.4.6.A8 Schutz vor unberechtigten oder manipulierenden Zugriffen auf das Dateisystem (S)

Wenn Zugriffe auf Dateien des SAP-Servers von Eingaben der Benutzenden abhängen, SOLLTEN diese Eingaben vor dem Zugriff validiert werden.

APP.4.6.A9 Berechtigungsprüfung in remote-fähigen Funktionsbausteinen (S)

Es SOLLTE sichergestellt werden, dass alle remote-fähigen Funktionsbausteine im Programmcode explizit prüfen, ob der Aufrufende berechtigt ist, die zugehörige Businesslogik auszuführen.

APP.4.6.A10 Verhinderung der Ausführung von Betriebssystemkommandos (S)

Jedem Aufruf eines erlaubten Betriebssystemkommandos SOLLTE eine entsprechende Berechtigungsprüfung (Berechtigungsobjekt *S_LOG_COM*) vorangestellt werden. Eingaben von Benutzenden SOLLTEN NICHT Teil eines Kom-

mandos sein. Deswegen SOLLTEN Betriebssystemaufrufe ausschließlich über dafür vorgesehene SAP-Standardfunktionsbausteine ausgeführt werden.

APP.4.6.A11 Vermeidung von eingeschleistem Schadcode (S)

Die ABAP-Befehle *INSERT REPORT* und *GENERATE SUBROUTINE POOL* SOLLTEN NICHT verwendet werden.

APP.4.6.A12 Vermeidung von generischer Modulausführung (S)

Transaktionen, Programme, Funktionsbausteine und Methoden SOLLTEN NICHT generisch ausführbar sein. Sollte es wichtige Gründe für eine generische Ausführung geben, SOLLTE detailliert dokumentiert werden, wo und warum dies geschieht. Zusätzlich SOLLTE eine Allowlist definiert werden, die alle erlaubten Module enthält. Bevor ein Modul aufgerufen wird, SOLLTE die Eingabe von Benutzenden mit der Allowlist abgeglichen werden.

APP.4.6.A13 Vermeidung von generischem Zugriff auf Tabelleninhalte (S)

Tabelleninhalte SOLLTEN NICHT generisch ausgelesen werden. Sollte es wichtige Gründe dafür geben, dies doch zu tun, SOLLTE detailliert dokumentiert werden, wo und warum dies geschieht. Außerdem SOLLTE dann gewährleistet sein, dass sich der dynamische Tabellename auf eine kontrollierbare Liste von Werten beschränkt.

APP.4.6.A14 Vermeidung von nativen SQL-Anweisungen (S)

Die Schnittstelle ABAP Database Connectivity (ADBC) SOLLTE NICHT verwendet werden. Eingaben von Benutzenden SOLLTEN NICHT Teil von ADBC-Befehlen sein.

APP.4.6.A15 Vermeidung von Datenlecks (S)

Es SOLLTE eine ausreichend sichere Berechtigungsprüfung durchgeführt werden, bevor geschäftskritische Daten angezeigt, übermittelt oder exportiert werden. Vorgesehene (gewollte) Möglichkeiten des Exports SOLLTEN dokumentiert werden.

APP.4.6.A16 Verzicht auf systemabhängige Funktionsausführung (S)

ABAP-Programme SOLLTEN NICHT systemabhängig programmiert werden, so dass sie nur auf einem bestimmten SAP-System ausgeführt werden können. Sollte dies jedoch unbedingt erforderlich sein, SOLLTE es detailliert dokumentiert werden. Außerdem SOLLTE der Code dann manuell überprüft werden.

APP.4.6.A17 Verzicht auf mandantenabhängige Funktionsausführung (S)

ABAP-Programme SOLLTEN NICHT mandantenabhängig programmiert werden, so dass sie nur von einem bestimmten Mandanten ausgeführt werden können. Sollte dies jedoch unbedingt erforderlich sein, SOLLTE es detailliert dokumentiert werden. Außerdem SOLLTEN dann zusätzliche Sicherheitsmaßnahmen ergriffen werden, wie beispielsweise eine manuelle Code-Überprüfung (manuelles Code-Review) oder eine Qualitätssicherung auf dem entsprechenden Mandanten.

APP.4.6.A18 Vermeidung von Open-SQL-Injection-Schwachstellen (S)

Dynamisches Open SQL SOLLTE NICHT verwendet werden. Falls Datenbankzugriffe mit dynamischen SQL-Bedingungen notwendig sind, SOLLTEN KEINE Eingaben von Benutzenden in der jeweiligen Abfrage übertragen werden. Wenn das dennoch der Fall ist, SOLLTEN die Eingaben von Benutzenden zwingend geprüft werden (Output Encoding).

APP.4.6.A19 Schutz vor Cross-Site-Scripting (S)

Auf selbst entwickeltes HTML in Business-Server-Pages-(BSP)-Anwendungen oder HTTP-Handlern SOLLTE möglichst verzichtet werden.

APP.4.6.A20 Keine Zugriffe auf Daten eines anderen Mandanten (S)

Die automatische Mandantentrennung SOLLTE NICHT umgangen werden. Auf Daten anderer Mandanten SOLLTE NICHT mittels *EXEC SQL* oder der Open SQL Option *CLIENT SPECIFIED* zugegriffen werden.

APP.4.6.A21 Verbot von verstecktem ABAP-Quelltext (S)

Der Quelltext eines selbst erstellten ABAP-Programms SOLLTE immer lesbar sein. Techniken, die das verhindern (Obfuscation), SOLLTEN NICHT verwendet werden.

3.3. Anforderungen bei erhöhtem Schutzbedarf

Im Folgenden sind für diesen Baustein exemplarische Vorschläge für Anforderungen aufgeführt, die über dasjenige Schutzniveau hinausgehen, das dem Stand der Technik entspricht. Die Vorschläge SOLLTEN bei erhöhtem Schutzbedarf in Betracht gezogen werden. Die konkrete Festlegung erfolgt im Rahmen einer individuellen Risikoanalyse.

APP.4.6.A22 Einsatz von ABAP-Codeanalyse Werkzeugen (H)

Zur automatisierten Überprüfung von ABAP-Code auf sicherheitsrelevante Programmierfehler, funktionale und technische Fehler sowie auf qualitative Schwachstellen SOLLTE ein ABAP-Codeanalyse-Werkzeug eingesetzt werden.

4. Weiterführende Informationen

4.1. Wissenswertes

Im „Best Practice Guide: Leitfaden Development ABAP 2.0“ der Deutschsprachigen SAP Anwendergruppe e.V. (DSAG) finden sich vertiefende Informationen zur ABAP-Programmierung.

Weitere Informationen und Best Practices zur sicheren ABAP-Programmierung finden sich im Buch „Sichere ABAP-Programmierung“ von Wiegenstein, Schuhmacher, Schinzel, Weidemann aus dem SAP Press Verlag.



APP.5.2 Microsoft Exchange und Outlook

1. Beschreibung

1.1. Einleitung

Microsoft Exchange Server (im Folgenden „Exchange“) ist eine Groupware-Lösung für mittlere bis große Institutionen. Mit ihr können elektronisch Nachrichten übermittelt werden und sie verfügt über weitere Dienste, um Workflows zu unterstützen. Nachrichten, wie E-Mails, können mit Exchange zentral verwaltet, zugestellt, gefiltert und versendet werden. Ebenso können typische Groupware-Anwendungen, wie Notizen, Kontaktlisten, Kalender und Aufgabenlisten angeboten und verwaltet werden. Um die Funktionen von Exchange nutzen zu können, ist neben dem Server-Dienst eine zusätzliche Client-Software oder ein Webbrowser nötig.

Microsoft Outlook (im Folgenden „Outlook“) ist ein Client für Exchange, der durch die Installation des Office-Pakets von Microsoft oder durch Integration in die Betriebssysteme von mobilen Geräten direkt zur Verfügung gestellt wird. Darüber hinaus ermöglicht die Webanwendung „Outlook im Web“ (ehemals „Outlook Web App“) über den Browser z. B. auf E-Mails, Kontakte und den Kalender zuzugreifen. Diese Funktion ist in Exchange bereits enthalten.

Die Kombination aus Exchange-Servern und Outlook-Clients wird in diesem Baustein als Exchange-System bezeichnet.

1.2. Zielsetzung

Das Ziel dieses Bausteins ist es, über typische Gefährdungen für Exchange und Outlook zu informieren sowie aufzuzeigen, wie Exchange und Outlook sicher in Institutionen eingesetzt werden.

1.3. Abgrenzung und Modellierung

Der Baustein ist auf alle Exchange-Systeme im Informationsverbund anzuwenden.

Allgemeine Anforderungen an die Sicherheit von E-Mail-Systemen sind im Baustein APP.5.3 *Allgemeiner E-Mail-Client und -Server* zu finden. Er ist zusätzlich auf jedes E-Mail-System anzuwenden, das auf Exchange bzw. Outlook basiert.

Der Baustein enthält spezifische Gefährdungen und Anforderungen für Exchange-Systeme. Spezifische Anforderungen an Serverplattformen und Betriebssysteme sind nicht Bestandteil des Bausteins. Diese sind in den Bausteinen SYS.1.1 *Allgemeiner Server* sowie SYS.2.1 *Allgemeiner Client* und in den jeweiligen betriebssystemspezifischen Bausteinen zu finden.

2. Gefährdungslage

Da IT-Grundschutz-Bausteine nicht auf individuelle Informationsverbünde eingehen können, werden zur Darstellung der Gefährdungslage typische Szenarien zugrunde gelegt. Die folgenden spezifischen Bedrohungen und Schwachstellen sind für den Baustein APP.5.2 *Microsoft Exchange und Outlook* von besonderer Bedeutung.

2.1. Fehlende oder unzureichende Regelungen für Exchange und Outlook

Übergreifende Regelungen und Vorgaben für Exchange und Outlook sind notwendig, damit die Sicherheit der Informationen, die mit Exchange und Outlook verarbeitet werden, gewährleistet wird. Beispielsweise können Daten verloren gehen, ungewollt verändert oder gelöscht werden, wenn Exchange fehlerhaft und ungeregelt in das Active Directory eingebunden wird. Ähnliches gilt, wenn Postfachdatenbanken ungeregelt depubliziert werden

und Exchange unzureichend in der Sicherheitsrichtlinie berücksichtigt wird. Gleches gilt, wenn die Outlook-Clients ungeregelt auf die Exchange-Server zugreifen können.

2.2. Fehlerhafte Migration von Exchange

Exchange-Systeme werden in der Praxis häufiger migriert als neu installiert. Um auf eine neue Version des Exchange Servers zu migrieren, muss in einigen Fällen das Betriebssystem auf eine neuere Version aktualisiert werden. Neue Versionen der Betriebssysteme stellen ihrerseits oft Anforderungen an das bestehende Domänenkonzept und die existierenden Verzeichnisdienste.

Wenn die Migration nicht sorgfältig geplant und durchgeführt wird, kann die interne Kommunikation über Exchange in der Institution massiv gestört werden, was in der Folge die Produktivität verringern könnte. Während der Migration können Probleme bei der Konfiguration auftreten, indem sich z. B. die Konfigurationseinstellungen für die unterschiedlichen Versionen oder die Möglichkeiten zur Anbindung an Verzeichnisdienste geändert haben. Des Weiteren können fehlerhafte Protokolleinstellungen zu Unregelmäßigkeiten bei der Informationsübermittlung, Authentisierung und Verschlüsselung führen.

2.3. Unzulässiger Browserzugriff auf Exchange

Mit Exchange können Anwendende über einem Browser auf das eigene E-Mail-Konto zugreifen. Hierzu werden die Internet Information Services (IIS) verwendet, die Bestandteil des Windows-Betriebssystems sind. Wenn diese Funktion unsachgemäß geplant und fehlerhaft konfiguriert wird, kann unter Umständen unkontrolliert von außen auf das interne Netz zugegriffen werden.

Wenn über das Internet mit einem Browser auf die E-Mails zugegriffen werden soll, birgt dies ein großes Gefahrenpotenzial. Ohne direkten Zugriff auf das Netz der Institution könnten Angreifende auf die E-Mails zugreifen und so unter anderem E-Mail-Adressen und -Inhalte ausspähen, E-Mail-Funktionen missbrauchen, Spam-Mails versenden sowie Zugang zu institutionsinternen Informationen erhalten.

2.4. Unerlaubte Anbindung anderer Systeme an Exchange

Exchange-Systeme sind eng mit dem Betriebssystem Windows verzahnt und arbeiten durch sogenannte Konnektoren (auch Connectors genannt) mit Fremdsystemen zusammen. Mithilfe der Konnektoren ist es anderen E-Mail-Systemen möglich, über bestimmte Protokolle (z. B. POP3) E-Mails von Exchange-Servern abzurufen.

Wenn bei der Installation oder einer Migration von Exchange die Konnektoren nicht mit berücksichtigt werden, können die vorhandenen Konnektoren inkompatibel zu der migrierten Exchange-Version sein. Hierdurch können E-Mails verloren gehen oder ungewollt verändert werden.

Außerhalb des homogenen Microsoft-Umfelds sind Sicherheitseinstellungen, die sich auf das Exchange-System beziehen, ungültig.

Wenn verschiedene Teilsysteme separat administriert werden, können stets Inkonsistenzen auftreten. Unsachgemäß angebundene Fremdsysteme können zudem zur Folge haben, dass Daten verloren gehen oder das Exchange-System blockiert wird.

2.5. Fehlerhafte Administration von Zugangs- und Zugriffsrechten unter Exchange und Outlook

Werden Zugangsrechte zu einem Outlook-Client bzw. auf innerhalb von Exchange und Outlook gespeicherte Daten fehlerhaft angelegt und administriert, können Sicherheitslücken entstehen. Dies ist beispielsweise der Fall, wenn über die notwendigen Rechte hinaus zusätzliche Rechte vergeben werden und dadurch unberechtigte Personen auf vertrauliche Informationen zugreifen können.

2.6. Fehlerhafte Konfiguration von Exchange

Eine häufige Ursache für erfolgreiche Angriffe auf Dienste wie Exchange sind fehlerhaft konfigurierte Exchange-Systeme. Da ein Exchange-System sehr komplex ist, können durch diverse Konfigurationseinstellungen und durch die sich gegenseitig beeinflussenden Parameter zahlreiche Sicherheitsprobleme entstehen. Die möglichen Fehlkonfigurationen erstrecken sich von der Installation und dem Betrieb der Exchange-Komponenten auf ungeeigneten IT-Systemen über nicht getätigte Verschlüsselungen und unzureichende Zugriffsbeschränkungen auf Exchange-Servern bis hin zur fehlerhaften Rechtevergabe bei der Erzeugung oder Initialisierung einer Exchange-Datenbank.

2.7. Fehlerhafte Konfiguration von Outlook

Der E-Mail-Client Outlook ist ein wichtiger Teil des Exchange-Systems. Für die Gesamtsicherheit des Exchange-Systems ist es wichtig, dass die Clients korrekt konfiguriert sind. Schon das ausgewählte Kommunikationsprotokoll kann spezielle Sicherheitsprobleme nach sich ziehen. Ebenso könnten private Schlüssel kompromittiert werden, mit denen E-Mails verschlüsselt und signiert werden. Wird auf Netzebene verschlüsselt, z. B. durch IPSec oder TLS, kann dieser Verschlüsselungsmechanismus bei einem fehlerhaft konfigurierten Client unwirksam werden. Durch Fehlkonfiguration können Sicherheitsprobleme entstehen, z. B. der Verlust der Vertraulichkeit durch unbefugten Zugriff.

2.8. Fehlfunktionen und Missbrauch selbst entwickelter Makros sowie Programmierschnittstellen unter Outlook

Viele Softwareherstellende sehen in ihren Tools und Anwendungen Programmierschnittstellen vor, sogenannte Application Programming Interfaces (APIs). Diese erlauben es, bestimmte Funktionen auch aus anderen Programmen heraus zu nutzen oder den Funktionsumfang der Anwendung zu erweitern. Solche Funktionen in Outlook können missbraucht werden, um Schadsoftware zu verbreiten. Zu den Schadsoftwarevarianten zählen z. B. bösartige Tools und Makros, die direkt Outlook und die damit verbundenen E-Mail-Funktionen ausnutzen, um Informationen abzugreifen, zu verändern oder zu löschen. Makros wiederum können dazu genutzt werden, Nachrichten, Termine oder Aufgaben weiterzuleiten oder zu verschieben. Dabei können Fehler in Makros ein erhöhtes Risiko darstellen. Indexfehler innerhalb von Makros können zu falschen Ergebnissen und zu möglicherweise unwirtschaftlichen Entscheidungen in der Institution führen. Spezifische Folgen können unnötige Kosten oder ein automatisierter Datenabfluss sein.

3. Anforderungen

Im Folgenden sind die spezifischen Anforderungen des Bausteins APP.5.2 *Microsoft Exchange und Outlook* aufgeführt. Der oder die Informationssicherheitsbeauftragte (ISB) ist dafür zuständig, dass alle Anforderungen gemäß dem festgelegten Sicherheitskonzept erfüllt und überprüft werden. Bei strategischen Entscheidungen ist der oder die ISB stets einzubeziehen.

Im IT-Grundschutz-Kompendium sind darüber hinaus weitere Rollen definiert. Sie sollten besetzt werden, insofern dies sinnvoll und angemessen ist.

Zuständigkeiten	Rollen
Grundsätzlich zuständig	IT-Betrieb
Weitere Zuständigkeiten	Keine

Genau eine Rolle sollte *Grundsätzlich zuständig* sein. Darüber hinaus kann es noch *Weitere Zuständigkeiten* geben. Falls eine dieser weiteren Rollen für die Erfüllung einer Anforderung vorrangig zuständig ist, dann wird diese Rolle hinter der Überschrift der Anforderung in eckigen Klammern aufgeführt. Die Verwendung des Singularen oder Plurals sagt nichts darüber aus, wie viele Personen diese Rollen ausfüllen sollen.

3.1. Basis-Anforderungen

Die folgenden Anforderungen MÜSSEN für diesen Baustein vorrangig erfüllt werden.

APP.5.2.A1 Planung des Einsatzes von Exchange und Outlook (B)

Bevor Exchange und Outlook eingesetzt werden, MUSS die Institution deren Einsatz sorgfältig planen. Dabei MUSS sie mindestens folgende Punkte beachten:

- Aufbau der E-Mail-Infrastruktur,
- einzubindende Clients beziehungsweise Server,
- Nutzung von funktionalen Erweiterungen sowie
- die zu verwendenden Protokolle.

APP.5.2.A2 Auswahl einer geeigneten Exchange-Infrastruktur (B)

Der IT-Betrieb MUSS auf Basis der Planung des Einsatzes von Exchange entscheiden, mit welchen IT-Systemen und Anwendungskomponenten sowie in welcher hierarchischen Abstufung die Exchange-Infrastruktur realisiert wird. Im Rahmen der Auswahl MUSS auch entschieden werden, ob die Exchange-Systeme als Cloud- oder lokaler Dienst betrieben werden sollen.

APP.5.2.A3 Berechtigungsmanagement und Zugriffsrechte (B)

Zusätzlich zum allgemeinen Berechtigungskonzept MUSS die Institution ein Berechtigungskonzept für die Systeme der Exchange-Infrastruktur erstellen, geeignet dokumentieren und anwenden.

Der IT-Betrieb MUSS serverseitige Benutzendenprofile für einen rechnerunabhängigen Zugriff der Benutzenden auf Exchange-Daten verwenden. Er MUSS die Standard-NTFS-Berechtigungen für das Exchange-Verzeichnis so anpassen, dass nur autorisierte Administrierende und Systemkonten auf die Daten in diesem Verzeichnis zugreifen können.

APP.5.2.A4 ENTFALLEN (B)

Diese Anforderung ist entfallen.

APP.5.2.A5 Datensicherung von Exchange (B)

Exchange-Server MÜSSEN vor Installationen und Konfigurationsänderungen sowie in zyklischen Abständen gesichert werden. Dabei MÜSSEN insbesondere die Exchange-Server-Datenbanken gesichert werden.

Gelöschte Exchange-Objekte SOLLTEN erst nach einiger Zeit aus der Datenbank entfernt werden.

3.2. Standard-Anforderungen

Gemeinsam mit den Basis-Anforderungen entsprechen die folgenden Anforderungen dem Stand der Technik für diesen Baustein. Sie SOLLTEN grundsätzlich erfüllt werden.

APP.5.2.A6 ENTFALLEN (S)

Diese Anforderung ist entfallen.

APP.5.2.A7 Migration von Exchange-Systemen (S)

Der IT-Betrieb SOLLTE alle Migrationsschritte gründlich planen und dokumentieren. Der IT-Betrieb SOLLTE dabei Postfächer, Objekte, Sicherheitsrichtlinien, Active-Directory-Konzepte sowie die Anbindung an andere E-Mail-Systeme berücksichtigen. Außerdem SOLLTE er Funktionsunterschiede zwischen verschiedenen Versionen von Exchange beachten. Das neue Exchange-System SOLLTE, bevor es installiert wird, in einem separaten Testnetz geprüft werden.

APP.5.2.A8 ENTFALLEN (S)

Diese Anforderung ist entfallen.

APP.5.2.A9 Sichere Konfiguration von Exchange-Servern (S)

Der IT-Betrieb SOLLTE Exchange-Server entsprechend der Vorgaben aus der Sicherheitsrichtlinie installieren und konfigurieren. Konnektoren SOLLTEN sicher konfiguriert werden. Der IT-Betrieb SOLLTE die Protokollierung des Exchange-Systems aktivieren. Für vorhandene benutzendenspezifische Anpassungen SOLLTE ein entsprechendes Konzept erstellt werden.

Bei der Verwendung von funktionalen Erweiterungen SOLLTE sichergestellt sein, dass die definierten Anforderungen an die Schutzziele Vertraulichkeit, Integrität und Verfügbarkeit weiterhin erfüllt sind.

APP.5.2.A10 Sichere Konfiguration von Outlook (S)

Der IT-Betrieb SOLLTE für jeden Benutzenden ein eigenes Outlook-Profil mit benutzendenspezifischen Einstellungen anlegen.

Der IT-Betrieb SOLLTE Outlook so konfigurieren, dass nur notwendige Informationen an andere Benutzende übermittelt werden. Der IT-Betrieb SOLLTE die Benutzenden darüber informieren, welche Informationen automatisiert

an andere Benutzende übermittelt werden. Lesebestätigungen und Informationen, die auf die interne Struktur der Institution schließen lassen, SOLLTEN NICHT an Externe übermittelt werden.

APP.5.2.A11 Absicherung der Kommunikation zwischen Exchange-Systemen (S)

Der IT-Betrieb SOLLTE nachvollziehbar entscheiden, mit welchen Schutzmechanismen die Kommunikation zwischen Exchange-Systemen abgesichert wird. Insbesondere SOLLTE der IT-Betrieb festlegen, wie die Kommunikation zu folgenden Schnittstellen abgesichert wird:

- Administrationsschnittstellen,
- Client-Server-Kommunikation,
- vorhandene Web-based-Distributed-Authoring-and-Versioning-(WebDAV)-Schnittstellen,
- Server-Server-Kommunikation und
- Public-Key-Infrastruktur, auf der die E-Mail-Verschlüsselung von Outlook basiert.

APP.5.2.A12 Einsatz von Outlook Anywhere, MAPI over HTTP und Outlook im Web (S)

Der IT-Betrieb SOLLTE Outlook Anywhere, MAPI over HTTP und Outlook im Web entsprechend den Sicherheitsanforderungen der Institution konfigurieren. Der Zugriff auf Exchange über das Internet SOLLTE auf die notwendigen Benutzenden beschränkt werden.

APP.5.2.A13 ENTFALLEN (S)

Diese Anforderung ist entfallen.

APP.5.2.A14 ENTFALLEN (S)

Diese Anforderung ist entfallen.

APP.5.2.A15 ENTFALLEN (S)

Diese Anforderung ist entfallen.

APP.5.2.A16 ENTFALLEN (S)

Diese Anforderung ist entfallen.

APP.5.2.A19 ENTFALLEN (S)

Diese Anforderung ist entfallen.

3.3. Anforderungen bei erhöhtem Schutzbedarf

Im Folgenden sind für diesen Baustein exemplarische Vorschläge für Anforderungen aufgeführt, die über dasjenige Schutzniveau hinausgehen, das dem Stand der Technik entspricht. Die Vorschläge SOLLTEN bei erhöhtem Schutzbedarf in Betracht gezogen werden. Die konkrete Festlegung erfolgt im Rahmen einer individuellen Risikoanalyse.

APP.5.2.A17 Verschlüsselung von Exchange-Datenbankdateien (H)

Der IT-Betrieb SOLLTE ein Konzept für die Verschlüsselung von PST-Dateien und Informationsspeicher-Dateien erstellen. Die Institution SOLLTE die Benutzenden über die Funktionsweise und die Schutzmechanismen bei der Verschlüsselung von PST-Dateien informieren. Weitere Aspekte für lokale PST-Dateien, die berücksichtigt werden SOLLTEN, wenn Exchange-Systemdatenbanken verschlüsselt werden, sind:

- eigene Verschlüsselungsfunktionen,
- Verschlüsselungsgrade sowie
- Mechanismen zur Absicherung der Daten in einer PST-Datei.

Mechanismen wie z. B. Encrypting File System oder Windows BitLocker Laufwerkverschlüsselung SOLLTEN zur Absicherung der PST-Dateien genutzt werden.

APP.5.2.A18 ENTFALLEN (H)

Diese Anforderung ist entfallen.

4. Weiterführende Informationen

4.1. Wissenswertes

Microsoft stellt auf seiner Webseite „Microsoft Technet“ (<https://technet.microsoft.com/de-de>) umfangreiche Informationen zur Administration von Microsoft Exchange zur Verfügung.



APP.5.3 Allgemeiner E-Mail-Client und -Server

1. Beschreibung

1.1. Einleitung

E-Mail ist eine der am häufigsten genutzten und ältesten Internetanwendungen. E-Mails werden dazu verwendet, Texte und angehängte Dateien zu versenden. Dazu wird eine E-Mail-Adresse benötigt.

Um E-Mail nutzen zu können, werden E-Mail-Server benötigt, die elektronische Nachrichten empfangen und versenden. In der Regel rufen E-Mail-Clients, mit denen auf E-Mail-Dienste zugegriffen wird, Nachrichten, die für sie bestimmt sind, mittels der Protokolle POP3 oder IMAP vom E-Mail-Server ab und senden mit dem Protokoll SMTP selbst Nachrichten an den E-Mail-Server, der diese bei Bedarf an einen anderen E-Mail-Server weiterleitet.

Da E-Mail insbesondere in Unternehmen und Behörden weit verbreitet ist, sind E-Mail-Server häufig das Ziel von Angriffen.

Auch E-Mail-Clients stehen im Fokus von Angriffen. Sie werden angegriffen, indem beispielsweise Schadsoftware per E-Mail versendet wird. Zusätzlich werden E-Mails auch oft als Werkzeug für Social-Engineering-Angriffe eingesetzt.

Aus diesen Gründen kommt dem sicheren Betrieb und der sicheren Nutzung von E-Mail-Anwendungen eine besondere Bedeutung zu.

1.2. Zielsetzung

Ziel dieses Bausteins ist es, die Informationen zu schützen, die mit E-Mail-Clients bzw. auf E-Mail-Servern verarbeitet werden.

1.3. Abgrenzung und Modellierung

Der Baustein APP.5.3 *Allgemeiner E-Mail-Client und -Server* ist auf jeden E-Mail-Client und -Server im Informationsverbund anzuwenden.

Der Baustein enthält Anforderungen für allgemeine E-Mail-Clients und -Server. Anforderungen für Serverplattformen, Betriebssysteme und Clients sind nicht Bestandteil des Bausteins. Diese sind in den Bausteinen SYS.1.1 *Allgemeiner Server* sowie SYS.2.1 *Allgemeiner Client* und in den jeweiligen betriebssystemspezifischen Bausteinen zu finden.

Der Baustein APP.5.3 *Allgemeiner E-Mail-Client und -Server* wird in einem Informationsverbund meist in Verbindung mit einem weiteren spezifischen Baustein der Schicht APP.5 *E-Mail/Groupware/Kommunikation* genutzt. Diese müssen ebenfalls separat umgesetzt werden. Zu diesen Bausteinen zählt unter anderem APP.5.2 *Microsoft Exchange und Outlook*.

Anforderungen für die Protokollierung und Datensicherung finden sich in den Bausteinen OPS.1.1.5 *Protokollierung* und CON.3 *Datensicherungskonzept*.

Nicht in diesem Baustein behandelt werden Groupware-Funktionen, die neben E-Mail auch noch weitere Funktionen wie die Verwaltung von Kontaktdaten und Kalendern bieten. Ebenso werden keine reinen Cloud-Lösungen behandelt, wie sie etwa als Teil von Microsoft 365 oder Google G Suite zu finden sind. Allgemeine Anforderungen dazu sind im Baustein OPS.2.2 *Cloud-Nutzung* zu finden. Ebenfalls nicht betrachtet werden Webbrowser, mit denen über Webseiten auf Webmail-Dienste zugegriffen wird. Anforderungen an Webbrowser sind im Baustein APP.1.2 *Webbrowser* zu finden.

2. Gefährdungslage

Da IT-Grundschutz-Bausteine nicht auf individuelle Informationsverbünde eingehen können, werden zur Darstellung der Gefährdungslage typische Szenarien zugrunde gelegt. Die folgenden spezifischen Bedrohungen und Schwachstellen sind für den Baustein APP.5.3 *Allgemeiner E-Mail-Client und -Server* von besonderer Bedeutung.

2.1. Unzureichende Planung der E-Mail-Nutzung

E-Mail-Anwendungen können ohne entsprechend dokumentierte Regelungen und ein definiertes Sicherheitsverfahren in der Institution nicht sicher genutzt werden. Falls in der Planung der E-Mail-Systeme die prozessualen, organisatorischen und technischen Regelungen vernachlässigt werden, könnte dies interne sowie externe Angriffe zur Folge haben.

Beispielsweise kann ein zu klein dimensionierter E-Mail-Server durch eine große Zahl von eingehenden E-Mails ausfallen. Werden keine ausreichenden Sicherheitsmaßnahmen geplant, ist es auch möglich, dass die E-Mail-Clients anfälliger für E-Mails sind, die Schadsoftware enthalten.

2.2. Fehlerhafte Einstellung von E-Mail-Clients und -Servern

Da eine E-Mail-Infrastruktur sehr komplex sein kann, können durch die vielen möglichen Einstellungen und durch die sich gegenseitig beeinflussenden Parameter zahlreiche Sicherheitsprobleme entstehen.

Beispielsweise kann ein E-Mail-Server durch eine fehlerhafte Konfiguration legitime E-Mails von anderen Servern ablehnen. Zusätzlich wäre es möglich, dass essenzielle Einstellungen ignoriert oder missachtet werden, z. B. die Transportverschlüsselung von E-Mails.

Außerdem kann eine falsche Konfiguration in E-Mail-Clients dazu führen, dass diese Schadcode in E-Mails ausführen. Diese Sicherheitslücken können zu einem signifikanten Verlust der Verfügbarkeit, Integrität und Vertraulichkeit von Informationen führen.

Viele Institutionen setzen keine Sicherheitsmechanismen ein, die es anderen E-Mail-Servern ermöglichen, zu überprüfen, ob eine E-Mail tatsächlich von der angegebenen Absendeadresse stammt. Außerdem kann ein falsch eingesetzter E-Mail-Server dazu missbraucht werden, um Spam-E-Mails zu versenden.

2.3. Unzuverlässigkeit von E-Mail

Über E-Mail-Anwendungen lassen sich schnell und komfortabel Daten austauschen. Das ist jedoch nicht immer zuverlässig. Zum Beispiel können durch fehlerhafte E-Mail-Server oder gestörte Übertragungswege Nachrichten verloren gehen. Ursachen dafür sind beispielsweise Spam-Filter, die legitime Nachrichten herausfiltern und verwerten. E-Mails können auch verloren gehen, wenn die Zieladresse nicht korrekt angegeben wurde. Im schlimmsten Fall können vertrauliche Informationen an falsche Zieladressen gesendet worden sein.

2.4. Schadsoftware in E-Mails

Es gibt verschiedene Wege, wie bei einem Angriff Schadsoftware mit Hilfe von E-Mails verbreitet werden kann. Einerseits kann schädlicher Code direkt in einer E-Mail enthalten sein. Ist der E-Mail-Client nicht richtig konfiguriert, wird der Code beim Öffnen der E-Mail ausgeführt.

Eine weitere Möglichkeit besteht darin, dass Dateien mit Schadsoftware als Anhang von E-Mails versendet werden. Falls solche E-Mails nicht durch Spam- oder Virenfilter aussortiert werden und die Anhänge geöffnet werden, wird die Schadsoftware ausgeführt. Diese kann zu weitreichenden Schäden auch für andere IT-Systeme führen, wenn beispielsweise Ransomware (oft als „Erpressungstrojaner“ bezeichnet) ausgeführt wird.

2.5. Social Engineering

E-Mails werden oft bei Angriffen dazu eingesetzt, um vertrauliche Informationen zu erhalten oder Personal zu anderem schädlichen Verhalten zu verleiten. Beispielsweise kann bei einem Angriff eine E-Mail gesendet werden, die vermeintlich von Vorgesetzten stammt und Anweisungen enthält, die der Institution schaden (sogenannter CEO-Fraud). Häufig wird dabei angewiesen, dass Geld auf Konten im Ausland überwiesen werden soll.

Möglich ist auch, dass eine gefälschte E-Mail einer eigentlich vertrauenswürdigen Quelle dazu auffordert, Zugangsdaten auf einer Webseite einzugeben (Phishing). Die so gewonnenen Zugangsdaten können dann bei einem Angriff für weitere Aktionen verwendet werden.

Verstärkt wird die Gefahr von Social Engineering, wenn die Institution nicht regelmäßig zu diesen Gefährdungen schult und sensibilisiert.

2.6. Mitlesen und Manipulieren von E-Mails

E-Mails werden in der Regel unverschlüsselt und ohne digitale Signatur versendet. Deswegen können bei einem Angriff E-Mails mitgelesen und sogar beliebig verändert werden. Auf diesem Weg können vertrauliche Informationen offengelegt oder falsche Informationen verteilt werden. Es ist auch möglich, dass auf diesem Weg Schadsoftware eingespielt wird.

3. Anforderungen

Im Folgenden sind die spezifischen Anforderungen des Bausteins APP.5.3 *Allgemeiner E-Mail-Client und -Server* aufgeführt. Der oder die Informationssicherheitsbeauftragte (ISB) ist dafür zuständig, dass alle Anforderungen gemäß dem festgelegten Sicherheitskonzept erfüllt und überprüft werden. Bei strategischen Entscheidungen ist der oder die ISB stets einzubeziehen.

Im IT-Grundschutz-Kompendium sind darüber hinaus weitere Rollen definiert. Sie sollten besetzt werden, insofern dies sinnvoll und angemessen ist.

Zuständigkeiten	Rollen
Grundsätzlich zuständig	IT-Betrieb
Weitere Zuständigkeiten	Benutzende, Vorgesetzte

Genau eine Rolle sollte *Grundsätzlich zuständig* sein. Darüber hinaus kann es noch *Weitere Zuständigkeiten* geben. Falls eine dieser weiteren Rollen für die Erfüllung einer Anforderung vorrangig zuständig ist, dann wird diese Rolle hinter der Überschrift der Anforderung in eckigen Klammern aufgeführt. Die Verwendung des Singularen oder Plurals sagt nichts darüber aus, wie viele Personen diese Rollen ausfüllen sollen.

3.1. Basis-Anforderungen

Die folgenden Anforderungen MÜSSEN für diesen Baustein vorrangig erfüllt werden.

APP.5.3.A1 Sichere Konfiguration der E-Mail-Clients (B)

Die Institution MUSS eine sichere Konfiguration für die E-Mail-Clients vorgeben. Die E-Mail-Clients MÜSSEN den Benutzenden vorkonfiguriert zur Verfügung gestellt werden.

Die Institution SOLLTE sicherstellen, dass sicherheitsrelevante Teile der Konfiguration nicht von Benutzenden geändert werden können. Ist dies nicht möglich, MUSS die Institution darauf hinweisen, dass die Konfiguration nicht selbstständig geändert werden darf.

Bevor Dateianhänge aus E-Mails geöffnet werden, MÜSSEN sie auf Schadsoftware überprüft werden. Die Dateianhänge MÜSSEN auf dem Client oder auf dem E-Mail-Server überprüft werden. E-Mail-Clients MÜSSEN so konfiguriert werden, dass sie eventuell vorhandenen HTML-Code und andere aktive Inhalte in E-Mails nicht automatisch interpretieren. Vorschaufunktionen für Datei-Anhänge MÜSSEN so konfiguriert werden, dass sie Dateien nicht automatisch interpretieren. E-Mail-Filterregeln sowie die automatische Weiterleitung von E-Mails MÜSSEN auf notwendige Anwendungsfälle beschränkt werden.

E-Mail-Clients MÜSSEN für die Kommunikation mit E-Mail-Servern über nicht vertrauenswürdige Netze eine sichere Transportverschlüsselung einsetzen.

APP.5.3.A2 Sicherer Betrieb von E-Mail-Servern (B)

Für den E-Mail-Empfang über nicht vertrauenswürdige Netze MÜSSEN E-Mail-Server eine sichere Transportverschlüsselung anbieten. Der Empfang von E-Mails über unverschlüsselte Verbindungen SOLLTE deaktiviert werden.

Versenden E-Mail-Server von sich aus E-Mails an andere E-Mail-Server, SOLLTEN sie eine sichere Transportverschlüsselung nutzen. Der IT-Betrieb SOLLTE den E-Mail-Versand durch unsichere Netze über unverschlüsselte Verbindungen deaktivieren.

Der IT-Betrieb MUSS den E-Mail-Server so konfigurieren, dass E-Mail-Clients nur über eine sichere Transportverschlüsselung auf Postfächer zugreifen können, wenn dies über nicht vertrauenswürdige Netze passiert.

Die Institution MUSS alle erlaubten E-Mail-Protokolle und Dienste festlegen. Der IT-Betrieb MUSS Schutzmechanismen gegen Denial-of-Service (DoS)-Attacken ergreifen. Werden Nachrichten auf einem E-Mail-Server gespeichert, MUSS der IT-Betrieb eine geeignete Größenbeschränkung für das serverseitige Postfach einrichten und dokumentieren. Außerdem MUSS der IT-Betrieb den E-Mail-Server so einstellen, dass er nicht als Spam-Relay missbraucht werden kann.

APP.5.3.A3 Datensicherung und Archivierung von E-Mails (B)

Der IT-Betrieb MUSS die Daten der E-Mail-Server und -Clients regelmäßig sichern. Dafür MUSS die Institution regeln, wie die gesendeten und empfangenen E-Mails der E-Mail-Clients sowie die E-Mails auf den Servern gesichert werden. Die Institution SOLLTE ebenfalls bei der Archivierung beachten, dass E-Mails möglicherweise nur lokal auf Clients gespeichert sind.

APP.5.3.A4 Spam- und Virenschutz auf dem E-Mail-Server (B)

Der IT-Betrieb MUSS sicherstellen, dass auf E-Mail-Servfern eingehende und ausgehende E-Mails, insbesondere deren Anhänge, auf Spam-Merkmale und schädliche Inhalte überprüft werden. Die Einführung und Nutzung von E-Mail-Filterprogrammen MUSS mit den Datenschutzbeauftragten und der Personalvertretung abgestimmt werden.

Die Institution MUSS festlegen, wie mit verschlüsselten E-Mails zu verfahren ist, wenn diese nicht durch das Virenschutzprogramm entschlüsselt werden können.

3.2. Standard-Anforderungen

Gemeinsam mit den Basis-Anforderungen entsprechen die folgenden Anforderungen dem Stand der Technik für diesen Baustein. Sie SOLLTEN grundsätzlich erfüllt werden.

APP.5.3.A5 Festlegung von Vertretungsregelungen bei E-Mail-Nutzung (S) [Vorgesetzte]

Die Institution SOLLTE Vertretungsregelungen für die Bearbeitung von E-Mails festlegen. Werden E-Mails weitergeleitet, SOLLTEN die Vertretenen mindestens darüber informiert werden. Bei der Weiterleitung von E-Mails MÜSSEN datenschutzrechtliche Aspekte berücksichtigt werden. Die Institution SOLLTE für Autoreply-Funktionen in E-Mail-Programmen Regelungen etablieren, die beschreiben, wie diese Funktionen sicher verwendet werden können. Wenn die Autoreply-Funktion benutzt wird, SOLLTEN keine internen Informationen weitergegeben werden.

APP.5.3.A6 Festlegung einer Sicherheitsrichtlinie für E-Mail (S)

Die Institution SOLLTE eine Sicherheitsrichtlinie für die Nutzung von E-Mails erstellen und regelmäßig aktualisieren. Die Institution SOLLTE alle Benutzenden und Administrierenden über neue oder veränderte Sicherheitsvorgaben für E-Mail-Anwendungen informieren. Die E-Mail-Sicherheitsrichtlinie SOLLTE konform zu den geltenden übergeordneten Sicherheitsrichtlinien der Institution sein. Die Institution SOLLTE prüfen, ob die Sicherheitsrichtlinie korrekt angewendet wird.

Die E-Mail-Sicherheitsrichtlinie für Benutzende SOLLTE vorgeben,

- welche Zugriffsrechte es gibt,
- wie E-Mails auf gefälschte Absendeadressen überprüft werden,
- wie sich übermittelte Informationen absichern lassen,
- wie die Integrität von E-Mails überprüft werden soll,
- welche offenen E-Mail-Verteiler verwendet werden dürfen,
- ob E-Mails privat genutzt werden dürfen,
- wie mit E-Mails und Postfächern ausscheidender Personen umgegangen werden soll,
- ob und wie Webmail-Dienste genutzt werden dürfen,